UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/687,301	10/15/2003	Russell Perry	300110121-2	7924

7590 07/02/2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collinsk, CO 80527-2400

EXAMINER	
EHICHIOYA, FRED I	

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
07/02/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/687,301	PERRY, RUSSELL
	Examiner	Art Unit
	Fred I. Ehichioya	2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 June 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 - 25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 - 25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on June 18, 2007 has been entered.

2. New claims 23 - 25 is added

3. Claims 1 – 25 are pending in this Office Action.

Response to Arguments

4. Applicant argues:

(a) Tatarinov does not disclose, teach or suggest at least "parsing each node ....., the attribute of that element (page 15, paragraph 2); (b) Tatarinov does not disclose, teach or suggest at parsing "a table having ...., the attribute of that element (page 16, paragraph 5 and (c) Tatarinov does not disclose, teach or suggest at least "reading data from ......the attribute of that element" (page 17, paragraph 4).

Examiner respectfully disagrees with the applicant. Since this argument is based on the new amendment, it will be shown in paragraph 6 below of this Office Action that

the combination of Tatarinov and DeGroote's reference discloses applicant's claimed limitations as argued.

Specification

5. The meaning of every term used in any of the claims should be apparent from the descriptive portion of the specification with clear disclosure as to its import; and in mechanical cases, it should be identified in the descriptive portion of the specification by reference to the drawing, designating the part or parts therein to which the term applies MPEP 608.01(o) [R-3].

The specification is objected because claims 1, 12 and 19 - 22 recite "constituent part(s)", "absolute" or "absolutely" and the specification fail to provide antecedent basis for the term "constituent part(s)", "absolute" and "absolutely".

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1 – 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over NPL "Storing and querying ordered XML using a relational database system" by Igor Tatarinov et al (hereinafter "Tatarinov") in view of U.S. Patent 7,076,763 issued to DeGroote et al (Hereinafter "DeGroote").

Regarding claim 1, Tatarinov discloses a method of storing a an XML document in a relational database (page 204 – Title and Abstract: "store and query XML documents using relational database"; "An XML document can be viewed as a tree/hierarchy" – page 205, section 3.1, paragraph 1) comprising:

(b) associating a unique identifier with a respective parsed node of the document (page 206, section 3.3: "the node in an XML document are assumed to have unique identifiers (IDs)") which identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: "each node is assigned a number that represents the node's absolute position in the document"); and

(c) storing each parsed constituent part of each node with its identifier in a table of a relational database (page 207 section 5.1: "the Edge table is used to store an entire document. . . .Each Edge tuple represents a node in the XML document tree"; Examiner submits that; Examiner submits that ***constituent part*** is neither defined by the specification nor the claims as shown in paragraph 5 above).

Though Tatarinov discloses parsing XML document as shown on page 214 section 7.8, Tatarinove does not explicitly discloses ***parsing each node of the XML*** as claimed.

However, DeGroote discloses (a) parsing each node of the XML document into constituent parts, including parsing elements and, where an element has an attribute, the attribute of that element (see column 11, lines 15 – 29 wherein each node of the XML is parsed and the attributes that are included in the elements of the XML are converted into XML node attributes).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "parsing each node of the XML" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 2, Tatarinov discloses a method according to claim 1, wherein the identifiers are associated such that a predetermined ordering of the identifiers and associated nodes in the database produces a predetermined ordering of nodes (page 206 section 3.3: "Accordingly, the result of evaluating an XPath expression is an ordered set of node IDs").

Regarding claim 3, Tatarinov discloses a method according to claim 2, wherein the predetermined ordering of the nodes is that produced by a depth first traversal of a tree representation of the hierarchical document (page 207 section 4.3: "each node is assigned a vector that represents the path from the document's root to the node. Each component of the path represents the local order of an ancestor node, as illustrated in Figure 1").

Regarding claim 4, Tatarinov discloses a method according to claim 1, wherein the identifier includes a separate character position for each hierarchical level in the document which is traversed to reach the associated node in the hierarchical document (page 208 section 5.1.1: "*Local Order:* Since the relative position of a node among its siblings does not uniquely identify a node in a document, unique node IDs still need to be assigned (that do not have to follow document order). A new column needs to be added to represent the position of a node among its siblings (the sibling index of a node, sIndex): Edge(id, parent_id, sIndex, path_id, value)".

Regarding claim 5, Tatarinov discloses a method according to claim 4, wherein a unique prefix character is used each time the number of nodes in a particular hierarchical level exceeds the unique characters in the identifier alphabet (page 211 section 6.2.2).

Regarding claim 6, Tatarinov discloses a method according to claim 1, wherein at least one database table entry includes a document identifier which identifies the hierarchical document from which an node has been parsed (page 207 section5.1, paragraph 2: "Each Edge tuple represents a node in the XML document tree. The id column corresponds to the node's ID and also serves as the primary key of the relation").

Regarding claim 7, Tatarinov discloses a method according to claim 1, wherein at least one database table entry includes a value field which records a value of the node in the table entry (page 207 section 5.1, paragraph 2, "value column is for text values of text nodes").

Regarding claim 8, Tatarinov discloses a method according to claim 1, wherein at least one database table entry includes a type field which indicates a characteristic type of the node in the table entry from a predetermined set of types (page 207 section 5.1: "A single relation, the Edge table is used to store an entire document. . . .Each Edge tuple represents a node in the XML document tree. The id column corresponds to the node's ID and also serves as the primary key of the relation").

Regarding claim 9, Tatarinov discloses a method according to claim 1, wherein the hierarchical document is an XML document (page 206 section 3.1, paragraph 1: "An XML document can be viewed as a tree/hierarchy").

Regarding claim 10, Tatarinov discloses a method according to claim 9, wherein at least one database table entry includes a type field which indicates a characteristic type of the node in the table entry from a predetermined set of types and wherein the set of types includes text node, element node, attribute node and/or processing instruction (page 207 section 5.1 paragraph 2: "values is used for text values of text nodes").

Regarding claim 11, Tatarinov discloses a method according to claim 9 or 10, wherein the database table includes YPath and ZPath indexes pointing to predetermined respective entries in respective node and ZPath database tables (Applicants disclose on the specification page 5, lines 11 – 13 "The NodePath refers to a unique element node in XML document. The NodePath can be split into two parts A/B/C/D and m/m/o/p, referred to as the YPath and ZPath respectively". Therefore, Examiner interprets "XPath expressions" disclosed on page 208 and Table 2 of Tatarinov as YPath and Zpath).

Regarding claim 12, Tatarinov discloses a relational database comprising

an identifier field for storing an identifier associated with each respective node stored in the node field (page 206 section 3.3: "the nodes in an input XML document are assumed to have unique identifiers (IDs)"), wherein the identifier identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: "each node is assigned a number that represents the node's absolute position in the document").

Though Tatarinov discloses a table having an node field for storing an node of a hierarchical document (page 206 section 4, paragraph 1: "In order to store and query shredded XML documents using a relational database system, we need some mechanism to capture document order in the relational data model. This is accomplished by encoding each node's position in an XML document as a data value").

Tatarinove does not explicitly disclose ***parsed constituent part of each node*** as claimed.

However, DeGroote discloses a table having an node field for storing each parsed constituent part of each node of an XML document including elements and, where an element has an attribute, the attribute of that element (see column 18, lines 6 – 8 wherein the variable table stores all the variable defined in the document. As shown in column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts. Examiner interprets these constituent parts as the variable stored in the variable table).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "parsed constituent parts" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 13, Tatarinov discloses a database according to claim 12, wherein at least one database table entry includes a document identifier field for storing a document identifier which identifies the hierarchical document from which an node has been parsed (page 207 section5.1, paragraph 2: "Each Edge tuple represents a node in the XML document tree. The id column corresponds to the node's ID and also serves as the primary key of the relation").

Regarding claim 14, Tatarinov discloses a database according to claim 12 or claim 13, wherein at least one database table entry includes a value field for recording a value of an node in the respective table entry (page 206, section 4.1 paragraph 1: "each node is assigned a number that represents the node's absolute position in the document").

Regarding claim 15,Tatarinov discloses a database according to any of claims 12 to 14, wherein at least one database table entry includes a type field for storing an indication of a characteristic type of an node in the respective table entry from a predetermined set of types (page 207 section 5.1: "A single relation, the Edge table is used to store an entire document. . . .Each Edge tuple represents a node in the XML document tree. The id column corresponds to the node's ID and also serves as the primary key of the relation").

Regarding claim 16, Tatarinov discloses a database according to any of claims 12 to 15, wherein the database table includes node and ZPath indexes referencing respective entries in respective node and ZPath database tables in the database (page 207section 5.1, paragraph 2: "Each Edge tuple represents a node in the XML document tree. The id column corresponds to the node's ID and also serves as the primary key of the relation. The parent_id column provides a "link" (i.e., foreign key) to the node's parent. The name column is used to store the tag name of element nodes, the value column is used for text values of text nodes").

Regarding claim 17, Tatarinov discloses a database according to claim 16, wherein the YPath table includes fields for storing XPath element names and document Ids (page 207 section 5.1 paragraph 2: The parent_id column . . . . tags name of element codes".

Regarding claim 18, Tatarinov discloses a database according to claim 16 or claim 17, wherein the ZPath table includes fields for storing XPath integer indexes and document Ids (page 205 section 3.2.1, paragraph 5: Also if predicate . . . the position of node selected").

Regarding claim 19, Tatarinov discloses a method of writing an XML document comprising:

(b) generating predetermined software events for respective read nodes (fig. 3: "XPath-toSQLtranslation algorithm" is interpreted as the "predetermined software events"), and

(c) passing the software events to a content handler which is arranged to translate each software event into a written node of the XML document (page 208 section 6.1, paragraph 2: "As shown, the algorithm in Figure 3 initially generates the SQL fragment to select the root elements of the stored XML documents (lines 4-6). Then, using the root elements as the initial context nodes, the algorithm generates the SQL fragments for each "step" of the XPath query being translated in order to produce new context nodes (line 8). The context nodes produced by the last step constitute the

query result (lines 10-11)"), each written node being associated with a unique identifier which identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: "each node is assigned a number that represents the node's absolute position in the document").

Though Tatarinov discloses reading data (page 205 section 3.2.1: examiner interprets "navigating" as "reading data") from a relational database (page 206 section 4, paragraph 1: 'XML documents using a relational database');

Tatarinove does not explicitly disclose ***constituent parts of each node of the XML document*** as claimed.

However, DeGroote discloses (a) reading data from a relational database which is representative of constituent parts of each node of the XML document, the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element (see column 11, lines 39 – 40 wherein XML nodes are read. As shown in column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts. Examiner interprets these constituent parts as the data read in the node).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "constituent parts of each node of the XML document" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined

by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 20, Tatarinov discloses a computer readable medium carrying a program which when executed on a computer causes storing an XML document in a relational database by: (page 204 – Title and Abstract: “store and query XML documents using relational database”; “An XML document can be viewed as a tree/hierarchy” – page 205, section 3.1, paragraph 1) comprising:

(b) associating a unique identifier with a respective parsed node of the document (page 206, section 3.3: “the node in an XML document are assumed to have unique identifiers (IDs)”) which identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: “each node is assigned a number that represents the node’s absolute position in the document”); and

(c) storing each parsed constituent part of each node with its identifier in a table of a relational database (page 207 section 5.1: “the Edge table is used to store an entire document. . . .Each Edge tuple represents a node in the XML document tree”; Examiner submits that; Examiner submits that ***constituent part*** is neither defined by the specification nor the claims as shown in paragraph 5 above).

Though Tatarinov discloses parsing XML document as shown on page 214 section 7.8; Tatarinove does not explicitly discloses ***parsing each node of the XML*** as claimed.

However, DeGroote discloses (a) parsing each node of the XML document into constituent parts, including parsing elements and, where an element has an attribute, the attribute of that element (see column 11, lines 15 – 29 wherein each node of the XML is parsed and the attributes that are included in the elements of the XML are converted into XML node attributes).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "parsing each node of the XML" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 21, Tatarinov discloses a computer readable medium carrying a program which when executed on a computer causes storing of a hierarchical document in a relational database by:

(a) receiving software events (page 207 section 7.2: "The rest of the queries were chosen to test key aspects of order-based functionality in XPath and XQuery. The test queries were translated to SQL using the algorithm described in Section 6") representing respective parsed nodes of the XML document (page 214, section 7.8, paragraph 1: XML document has to be parsed"),

(b) associating a unique identifier with respective parsed nodes of the document (page 206, section 3.3: "the node in an XML document are assumed to have unique

identifiers (IDs)") which identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: "each node is assigned a number that represents the node's absolute position in the document").

Though Tatarinov discloses storing the node with its identifier in a table of a relational database (page 207 section 5.1: "the Edge table is used to store an entire document. . . .Each Edge tuple represents a node in the XML document tree"); Tatarinove does not explicitly discloses ***constituent parts of each node of the document*** claimed.

However, DeGroote discloses (c) storing the constituent parts of each node of the document with its identifier in a table of a relational database (see column 11, lines 35 – 37 wherein the XML objects are stored), the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element (see column 11, lines 15 – 29 wherein each node of the XML is parsed and the attributes that are included in the elements of the XML are converted into XML node attributes).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "constituent parts of each node of the document" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 22, Tatarinov discloses a computer readable medium carrying a program which when executed on a computer causing writing of an XML document by:

(b) generating predetermined software events for respective read nodes (fig. 3: "XPath-toSQLtranslation algorithm" is interpreted as the "predetermined software events"), and

(c) passing the software events to a content handler which is arranged to translate each software event into a written node of the XML document (page 208 section 6.1, paragraph 2: "As shown, the algorithm in Figure 3 initially generates the SQL fragment to select the root elements of the stored XML documents (lines 4-6). Then, using the root elements as the initial context nodes, the algorithm generates the SQL fragments for each "step" of the XPath query being translated in order to produce new context nodes (line 8). The context nodes produced by the last step constitute the query result (lines 10-11)"), each written node being associated with a unique identifier which identifies, absolutely, the hierarchical position of the node in the document (page 206 section 4.1: "each node is assigned a number that represents the node's absolute position in the document").

Though Tatarinov discloses reading data (page 205 section 3.2.1: examiner interprets "navigating" as "reading data") from a relational database (page 206 section 4, paragraph 1: 'XML documents using a relational database');

Tatarinove does not explicitly disclose ***constituent parts of each node of the XML document*** as claimed.

However, DeGroote discloses (a) reading data from a relational database which is representative of constituent parts of each node of the XML document, the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element (see column 11, lines 39 – 40 wherein XML nodes are read. As shown in column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts. Examiner interprets these constituent parts as the data read in the node).

It would have been obvious to one of ordinary skills at the data processing art at the time of present invention to combine the cited references, because DeGroote's teaching of "constituent parts of each node of the XML document" would have allowed Tatarinov's system to scale to a minimum the component need for application development. Only the parts that are needed in the parsed nodes that are determined by the XML tags are included the component for application development thereby reducing unnecessary resources.

Regarding claim 23, DeGroote discloses the method of claim 1 further comprising:

reading data from the relational database which is representative of each node of the XML document (see column 11, lines 39 – 40 wherein XML nodes are read); and

writing the data into a document (see column 11, lines 25 – 25 wherein element objects are then included in the document) such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical

positioning as indicated by the unique identifier for each node (see column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts). t is inherent that every node has internal key/identifier that identifies a particular node).

Regarding claim 24, DeGroote discloses the computer readable medium of claim 20, the program causing the computer to:

read data from the relational database which is representative of each node of the XML document (see column 11, lines 39 – 40 wherein XML nodes are read); and

write the data into a document such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical positioning as indicated by the unique identifier for each node (see column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts. It is inherent that every node has internal key/identifier that identifies a particular node).

Regarding claim 25, DeGroote discloses the computer readable medium of claim 21, the program causing the computer to:

read data from the relational database which is representative of each node of the XML document (see column 11, lines 39 – 40 wherein XML nodes are read); and

write the data into a document such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical

positioning as indicated by the unique identifier for each node (see column 11, lines 15 – 29 wherein each node of the XML is parsed into the elements and attributes that are the constituent parts. It is inherent that every node has internal key/identifier that identifies a particular node).

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Fred I. Ehichioya whose telephone number is 571-272-4034. The examiner can normally be reached on M - F 8:00 AM to 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on 571-272-4107. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Fred I. Ehichioya
Patent Examiner
Art Unit 2162

June 22, 2007